Stereo matching based on nonlinear diffusion with disparity-dependent support weights

K.-J. Yoon

School of Information and Communications, Computer vision Lab., GIST, Korea
E-mail: kjoyoon@gist.ac.kr

Abstract: In stereo matching, computing matching cost or similarity between pixels across different images is one of the main steps to get reliable results. More accurate and robust matching cost can be obtained by aggregating per-pixel raw matching cost within the predefined support area. Here, it is very important to aggregate only valid supports from neighbouring pixels. However, unfortunately, it is hard to evaluate the validity of the supports from neighbours beforehand. To resolve this problem, we propose a new method for the matching cost computation based on the nonlinear diffusion. The proposed method helps to aggregate truly valid supports from neighbouring pixels and does not require any local stopping criterion of iteration. This is achieved by using disparity-dependent support weights that are also updated at every iteration. As a result, the proposed method combined with a simple winner-take-all disparity selection method yields good results not only in homogeneous areas but also in depth discontinuity areas as the iteration goes on without the critical degradation of performance. In addition, when combined with global methods for the disparity selection, the proposed method truly improve the matching performance.

1 Introduction

Nowadays, stereo matching is commonly used to extract three-dimensional information of a scene using images in many systems. Many methods have been proposed for decades to solve the stereo matching problem, and they can be roughly divided into two categories according to disparity selection methods as in [1]: local methods and global methods. Local methods [2–5] select disparities of pixels locally by using the winner-take-all (WTA) method. They typically use some kinds of statistical correlation among colour or intensity patterns in local support areas. In contrast, global methods [6–10] seek a disparity surface minimising the global cost function defined by making an explicit smoothness assumption. Most of them mainly focus on how to efficiently minimise conventional matching costs, in spite of the fact that lower cost solutions do not always correspond to better performance as pointed out in [11] and [12]. Therefore defining and computing matching cost or similarity between pixels is one of the main steps in both local and global methods.

In general, more accurate and robust matching cost can be obtained by aggregating per-pixel raw matching costs within the predefined support area. Supports are aggregated locally by summing over a support area or by diffusion in order to increase robustness and to decrease image ambiguity. In this approach, it is very important to aggregate only valid supports from neighbouring pixels. Therefore the performance of this approach is highly dependent on the shape/size of a support area. Ideally, the local support area should include all and only those neighbouring pixels that are come from the same depth. However, because we do not know the depths of pixels beforehand, it is very difficult to find an optimal window for each pixel. In this context, we proposed a new method in [4], which is based on adaptive support weights. However, we should use very large local support windows to deal with homogeneous areas because we do not know the sizes of homogeneous regions beforehand – the size of local support windows should be larger than those of the homogeneous regions – and, therefore huge memory and computation power are required.

Instead of using a fixed-sized local support window, the support aggregation can be achieved via diffusion. Here, the problem with this approach is that the size of the support area increases as the iteration goes on. More aggregation (larger number of iterations) clearly helps to recover textureless areas. However, at the same time, too much aggregation causes errors near depth discontinuities. Therefore the number of iterations and some local stopping criteria are very important factors in this approach, as the size/shape of a local support window in the window-based approach.

In this paper, we propose a new nonlinear diffusion method for the support aggregation, aiming at yielding good results not only in homogeneous areas but also in depth discontinuity areas. The proposed method is essentially based on disparity-dependent support weights. Supports are iteratively aggregated with the support weights while adjusting the support weights according to disparities to prevent ambiguous supports from neighbouring pixels.

This work can be thought as the extension of our previous works [4, 12, 13], which consider the reference and the target images together in the support aggregation. To prevent...
erroneous supports coming from different depths, most previous works make use of colour or intensity gradient, assuming that depth discontinuities coincide with colour or intensity edges. This assumption is generally valid, but previous methods consider only the reference image and this cannot help to prevent supports from occluded pixels as shown in Fig. 1. Even though pixels in A and B have the same colour, we have to prevent the supports coming from B to A because pixels in B are not visible in the target image.

2 Modelling of support weights

We first need to build the model for support-weight assignment. Needless to say, the support from a neighbouring pixel is valid only when the neighbouring pixel is ‘from the same depth’ with the pixel under consideration and ‘not occluded’ (i.e. visible in both images). Therefore the support weight of a neighbouring pixel should be in proportion to the probability that it is from the same depth and it is visible in both images. When \( p \) is the pixel under consideration and \( q \) is a pixel in the support area of \( p \), \( N_p \), the simple model can be expressed as

\[
w(q, p) = \Pr(O_{pq} | O_p^q) = \Pr(O_{pq}) \Pr(O_p^q)
\]

(1)

when assuming two events \( O_{pq} \) and \( O_p^q \) are independent. Here, \( w(q, p) \) represents the support weight of \( q \) and \( O_{pq} \) represents the event that \( p \) and \( q \) are from the same depth. \( O_p^q \) is the event that \( q \) is visible in both images. Here, by introducing the binary function \( v(q) \) representing the visibility of \( q \) as

\[
v(q) = \begin{cases} 1 & \text{if } q \text{ is visible in both images} \\ 0 & \text{otherwise} \end{cases}
\]

(2)

\( O_p^q \) represents the event \( \{v(q) = 1\} \).

However, this modelling is for only the reference image and does not consider the depth (i.e. disparity) under consideration. In our work, we extend the model for considering both images and the disparity. In this case, our model can be expressed as

\[
w(q, p, d) = \Pr(O_{pq} | O_p^q) \Pr(O_p^q | O_q^d) \Pr(O_q^d)
\]

(3)

where \( w(q, p, d) \) is the support weight of \( q \) for a disparity \( d \). Here, \( w(q, p, d) = w(q, p, d) = w(q, p, d) \). \( \bar{p}_d \) and \( \bar{q}_d \) are the corresponding pixels in the target image when \( p \) and \( q \) in the reference image have a disparity \( d \). [It is assumed without loss of generality that images are rectified so that the disparities of pixels are restricted to 1D scalar values.] \( O_p^q \) represents the event that the disparity of \( p \) is \( d \) as \( \{d_p = d\} \).

Similarly, \( O_p^q \) and \( O_q^d \) represent the events \( \{d_p = d\} \) and \( \{d_q = d\} \), respectively. When \( p \) in the target image corresponds to \( \bar{p}_d \) in the target image with a disparity \( d \), we also define the disparity of \( \bar{p}_d \) as \( d \).

The support-weight modelling is then specified as the modelling of the probabilities, \( \Pr(O_{pq}^d | O_p^q) \) and \( \Pr(O_q^d | O_p^q) \), and \( \Pr(O_q^d) \), as shown in (3). \( \Pr(O_{pq}^d | O_p^q) \) and \( \Pr(O_q^d | O_p^q) \) are the probabilities that neighbouring pixels have the same disparity (i.e. from the same depth) in each image. In fact, this model is almost similar to the model proposed in [4] except \( \Pr(O_q^d) \). By using \( \Pr(O_q^d) \), we can consider the supports coming from occluded pixels more effectively. However, the problem is that we do not know the disparities and visibility of pixels beforehand because they are what we want to compute. For this reason, some methods [2, 14] iteratively update support windows or support weights. The iterative methods, however, are very sensitive to the initial disparity estimation. To resolve this dilemma, we need some reasonable constraints or assumptions. In this work, we assume that depth discontinuities generally coincide with colour or intensity edges. In addition, we assume that if a neighbouring pixel is from the same depth and visible in both images, similar colours are observed at the same position in both images. We develop an efficient method based on these assumptions.

3 Support-weight computation

According to (3), we should assign small support weights to pixels that are probably coming from different depths or occluded regions. To this end, we first compute adaptive support weights based on the colour similarity and the distance between pixels in each image to block the supports coming from different depths as we proposed in [4], and then recompute the support weights used for support aggregation according to disparities to block the supports coming from occluded areas. In Section 3.1, we will briefly review the adaptive support-weight computation proposed in [4], and then the support-weight updating scheme will be introduced in Section 3.2.

3.1 Adaptive support-weight approach

The adaptive support weights are computed based on the colour similarity and the proximity between the pixel under consideration and neighbouring pixels. The more similar the colour of a pixel, the larger its support weight. In addition, the closer the pixel is, the larger the support weight. The support weight of a neighbouring pixel is defined using the Laplacian kernel as

\[
w_l(p, q) = \exp \left( -\frac{\Delta c_{pq}}{\gamma_c} - \frac{\Delta g_{pq}}{\gamma_g} \right)
\]

(4)

where \( \Delta c_{pq} \) and \( \Delta g_{pq} \) represent the colour difference and the distance between \( p \) and \( q \) in an image, respectively. \( \gamma_c \) and \( \gamma_g \) are control parameters. In the same manner, we can compute \( w_l(\bar{p}_d, \bar{q}_d) \) as

\[
w_l(\bar{p}_d, \bar{q}_d) = \exp \left( -\frac{\Delta c_{\bar{p}_d \bar{q}_d}}{\gamma_c} - \frac{\Delta g_{\bar{p}_d \bar{q}_d}}{\gamma_g} \right)
\]

(5)

These support weights encourage the pixel having the high probability of the same depth with the pixel of interest. Therefore \( w_l(p, q) \) and \( w_l(\bar{p}_d, \bar{q}_d) \) are related to \( \Pr(O_{pq}^d | O_p^q) \) and \( \Pr(O_{\bar{q}_d}^d | O_{\bar{p}_d}^q) \), respectively. Here, these

![Fig. 1 Depth discontinuity area](image-url)

Supports from different depths and occluded pixels in B to pixels in A may cause false matches.
support weights are entirely based on the contextual information within given support areas and do not depend on the initial disparity estimation at all.

For reducing the effect of half-occluded pixels (e.g. pixels in B in Fig. 1), support weights are then combined as the function of disparities to be considered. The support weight of a neighbouring pixel q in Np, for the disparity d is computed by combining two support weights in both images as

\[ w(p, q, d) = w_r(p, q) \times w_a(q_g, q_d) \]
\[ = \exp\left(-\frac{\Delta c_{pq} + \Delta d_{pq}}{\gamma_d}\right) \times \exp\left(-\frac{\Delta c_{pq} + \Delta d_{pq}}{\gamma_r}\right) \]  \hspace{1cm} (6)

Here, \( w(p, q, d) \) is equal to \( w(q, p, d) \) and these support weights do not change during the whole process.

### 3.2 Support weights for occluded pixels

The blocking of supports coming from occluded pixels is clearly shown in Fig. 2. Figs. 2a and c show two local support windows and Figs. 2b and d show support weights corresponding to Figs. 2a and c computed by (4) and (5), respectively. Here, the pixel A in Fig. 2a is corresponding to the pixel A in Fig. 2e with the disparity \( d_A \). In contrast, the pixel B in Fig. 2a is not corresponding to the pixel B’ in Fig. 2e for \( d_B \) because of occlusion. Therefore the support coming from B to A should be suppressed during the aggregation step when considering the disparity \( d_A \). However, if we consider the reference image only and use the adaptive support weights shown in Fig. 2b, the pixel A may get the incorrect support from B (because two pixels have almost the same colour) and this may result in a false match. However, if we adjust support weights according to the considered disparity as shown in Fig. 2e, we can effectively block the support coming from B – we can see that the pixel B actually has a very small support weight in Fig. 2e.

From Fig. 2, we can see that the disparity-dependent support weights fairly block the supports coming from occluded pixels. However, it is impossible to perfectly block incorrect supports as the iteration goes on. This is clear because the model we use is not perfect. To complement this, we detect the pixels that are likely to be occluded at each iteration and consider detected pixels in the support aggregation. Those pixels can be detected by thresholding the lowest aggregated cost as in [10] or by the left–right consistency check after applying the WTA method to both images. Once the pixel is detected as a potentially occluded pixel at the \( (n-1) \)th iteration, the support weight of the pixel at the \( n \)th iteration is given by using the binary function \( v(\cdot) \) defined in (2) as

\[ w'(q) = \exp\left(-\frac{1 - v(q)}{\gamma_v}\right) \]  \hspace{1cm} (7)

where \( \gamma_v \) is a control parameter.

### 4 Support aggregation via diffusion

#### 4.1 Nonlinear diffusion of supports

The nonlinear diffusion for support aggregation can be simply achieved by using the disparity-dependent support weights and the visibility function. Supports are iteratively aggregated as

\[ S_n(p, d) = \frac{1}{z} \sum_{q \in \mathcal{N}_p} w(p, q, d)w_{n-1}(q)S_{n-1}(q, d) \]  \hspace{1cm} (8)

where \( z \) is the normalisation constant as \( z = \sum_{q \in \mathcal{N}_p} w(p, q, d)w_{n-1}(q) \) and \( n \) denotes the iteration number. Initially, \( w_0(\cdot) = 1 \) for all pixels. \( S_0(p, d) \) is the initial matching cost of \( p \) for disparity \( d \), which can be computed using any function of image colours.

#### 4.2 Stereo matching using the proposed diffusion

In this section, we propose a simple stereo method based on the proposed nonlinear diffusion of supports. Firstly, the
**Fig. 5** Dense disparity maps for the ‘Tsukuba’ and ‘Venus’ images

- **a** Left images
- **b** Ground truth
- **c** Bay diff [8]
- **d** Stoch. diff. [16]
- **e** Our results

**Fig. 6** Dense disparity maps for the new testbed images

- **a** Left images
- **b** Ground truth
- **c** BP + directedDiff [17]
- **d** PhaseDiff [19]
- **e** AdaptWeight [4]
- **f** Our results
initial per-pixel raw matching costs are computed by using the truncated absolute difference (AD) as

$$S_0(p, d) = \min \left\{ \sum_{c \in \{r, g, b\}} |I_c(p) - I_c(\overline{p}, d)|, T \right\}$$  (9)

where $I_c$ is the intensity of the colour band $c$ and $T$ is the truncation value that controls the limit of the matching cost. Using the initial per-pixel raw matching costs, the support aggregation is achieved by the proposed method. In addition, for the fair verification of the proposed support-aggregation method, we do not perform any complicated optimisation, but we adopt the simplest WTA method that selects the disparity of each pixel locally. The disparity with the lowest aggregated cost is selected as

$$d_p = \arg \min_{d \in C_d} S(p, d)$$  (10)

where $C_d = \{d_{\text{min}}, \ldots, d_{\text{max}}\}$ is the set of all possible disparities. This WTA is followed by a left–right consistency check for invalidating occlusions and mismatches. Invalid disparity areas are then filled by propagating neighbouring disparities.

Fig. 7 Dense disparity maps for the other image sets

a Left images
b Right images
c Ground truth
d Our results
small (i.e. background) disparity values as in [15] where they evaluate and compare cost functions for stereo matching. We also apply the median filter to resultant disparity maps to remove salt-and-pepper errors, because the support-weight computation using a single pixel colour is sensitive to image sampling and image noise. The reason we perform these post-processing steps, as opposed to comparing raw results, is to reduce overall errors as performed in [15].

5 Experiments

We verify the efficiency of the proposed method by using the real images with ground truth, which are often used for performance comparison of various methods as in [1]. The proposed methods are run with a constant parameter setting across all images – the size of a local support window is fixed as $(5 \times 5)$ and the number of iterations is set 300 that is fully enough for the convergence for all images although only 50–60 iterations are enough for convergence in practice. In addition, we set $T = 40$, $\gamma_c = 5$, $\gamma_p = 2.5$ and $\gamma_v = 1$.

Figs. 3 and 4 show the performance of the proposed method for ‘Tsukuba’ images with respect to the iteration number. Unlike other support aggregation methods where the errors in depth discontinuity areas terribly increase as the iteration goes on (or as the size of the support area increases), the performance of the proposed method in textureless areas and in depth discontinuity areas remains almost constant without critical degradation after the small number of iterations. Therefore the number of iteration is

![BP results with/without the proposed method](image)

**Fig. 8** BP results with/without the proposed method

From left to right, results of the proposed method, results of the BP method, and the results of the combined (proposed measure + BP) method. The numbers under each result represent the percentage of bad pixels ($e > 1$) in nonoccluded, all, and depth discontinuity areas, respectively

- $a$ (1.67, 2.59, 7.5)
- $b$ (2.31, 3.01, 9.77)
- $c$ (1.34, 2.16, 6.93)
- $d$ (5.47, 11.3, 10.0)
- $e$ (29.7, 33.4, 34.3)
- $f$ (4.95, 10.8, 10.1)

![Same experimental results as in Fig. 8 with other data sets provided by [18]](image)

**Fig. 9** Same experimental results as in Fig. 8 with other data sets provided by [18]

- $a$ Left image
- $b$ Right image
- $c$ Ground truth
- $d$ Proposed (3.25, 4.12, 1.37)
- $e$ BP only (7.12, 9.45, 4.71)
- $f$ Proposed + BP (3.21, 3.99, 1.60)
Performance of the proposed method compared with other diffusion-based methods and some other local methods for new testbed images

<table>
<thead>
<tr>
<th></th>
<th>Tsukuba</th>
<th></th>
<th>Venus</th>
<th></th>
<th>Teddy</th>
<th></th>
<th>Cone</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nonocc</td>
<td>untex</td>
<td>disc</td>
<td>nonocc</td>
<td>untex</td>
<td>disc</td>
<td>nonocc</td>
<td>untex</td>
</tr>
<tr>
<td>proposed</td>
<td>1.14</td>
<td>0.26</td>
<td>6.12</td>
<td>1.15</td>
<td>0.02</td>
<td>5.90</td>
<td>0.73</td>
<td>0.46</td>
</tr>
<tr>
<td>Bay. diff [8]</td>
<td>6.49</td>
<td>11.62</td>
<td>12.29</td>
<td>1.45</td>
<td>0.72</td>
<td>9.29</td>
<td>4.00</td>
<td>7.21</td>
</tr>
<tr>
<td>Stoch. diff. [16]</td>
<td>3.95</td>
<td>4.08</td>
<td>15.49</td>
<td>2.45</td>
<td>0.90</td>
<td>10.58</td>
<td>2.45</td>
<td>2.41</td>
</tr>
</tbody>
</table>

Table 2: Performance of the proposed method compared with other diffusion-based methods and some other local methods for new testbed images

<table>
<thead>
<tr>
<th></th>
<th>Tsukuba</th>
<th></th>
<th>Venus</th>
<th></th>
<th>Teddy</th>
<th></th>
<th>Cone</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nonocc</td>
<td>all</td>
<td>disc</td>
<td>nonocc</td>
<td>all</td>
<td>disc</td>
<td>nonocc</td>
<td>all</td>
</tr>
<tr>
<td>GeoSup [5]</td>
<td>1.45</td>
<td>1.83</td>
<td>7.71</td>
<td>0.14</td>
<td>0.26</td>
<td>1.90</td>
<td>6.88</td>
<td>13.2</td>
</tr>
<tr>
<td>DistinctSM [13]</td>
<td>1.21</td>
<td>1.75</td>
<td>6.39</td>
<td>0.35</td>
<td>0.69</td>
<td>2.63</td>
<td>7.45</td>
<td>13.0</td>
</tr>
<tr>
<td>AdaptWeight [4]</td>
<td>1.38</td>
<td>1.85</td>
<td>6.90</td>
<td>0.71</td>
<td>1.19</td>
<td>6.13</td>
<td>7.88</td>
<td>13.3</td>
</tr>
<tr>
<td>BP + DirectedDiff [17]</td>
<td>2.90</td>
<td>4.47</td>
<td>15.1</td>
<td>0.65</td>
<td>1.20</td>
<td>4.52</td>
<td>5.07</td>
<td>14.7</td>
</tr>
<tr>
<td>DCBGrid [18]</td>
<td>5.90</td>
<td>7.26</td>
<td>21.0</td>
<td>1.35</td>
<td>1.91</td>
<td>11.2</td>
<td>10.5</td>
<td>17.2</td>
</tr>
<tr>
<td>PhaseDiff [19]</td>
<td>4.89</td>
<td>7.11</td>
<td>16.3</td>
<td>8.34</td>
<td>9.76</td>
<td>26.0</td>
<td>20.0</td>
<td>28.0</td>
</tr>
<tr>
<td>SSD + MF [20]</td>
<td>5.23</td>
<td>7.07</td>
<td>24.1</td>
<td>3.74</td>
<td>5.16</td>
<td>11.9</td>
<td>16.5</td>
<td>24.8</td>
</tr>
<tr>
<td>proposed</td>
<td>1.14</td>
<td>1.93</td>
<td>6.12</td>
<td>0.73</td>
<td>1.21</td>
<td>3.81</td>
<td>8.02</td>
<td>14.2</td>
</tr>
</tbody>
</table>

not so critical and no local stopping criteria is needed in the proposed method. This is the one of main advantages of the proposed method.

The results for old and new testbed images are given in Figs. 5 and 6. As shown in Fig. 6, the proposed method yields accurate results in all images areas (including depth discontinuity areas) for all testbed images. More results using other images are also given in Fig. 7. The performance is summarised in Tables 1 and 2 to compare the performance with other state-of-the-art local and diffusion-based methods. The numbers in Tables 1 and 2 represent the percentage of bad pixels (i.e. pixel whose absolute disparity error is greater than 1). We can see that the proposed method yields comparable or better results even though we just use the simple WTA method without any complicated processes. Especially, the performance in depth discontinuity areas is much better than those of other diffusion-based methods. This shows the effectiveness of the proposed support aggregation method.

As mentioned, the proposed method can be easily applied to global stereo methods without much modification. To verify the performance improvement by the proposed method, we applied the proposed method to the belief-propagation (BP) method implemented by Tappen in [11]. The BP method was run with a arbitrarily selected constant parameter setting across all images. [The arbitrarily selected parameters do not matter in this experiment because what we want to do in this experiment is not to obtain the best performance but to verify the performance improvement.] The results obtained by combining the proposed method with other global methods are also shown in Figs. 8 and 9. Here, we set $T = 40$, $\gamma_c = 2$, $\gamma_p = 5$, $\gamma_v = 2$, and the number of iterations 30 for the proposed method to clearly see the performance enhancement. From the result, we can see that the proposed method truly improves the performance of global methods – it can be used for any global methods as the matching cost computation step.

Here, it is worthy of notice that the overall characteristics and performance of the proposed method are similar to those of the adaptive support-weight approach [4] because the proposed method is also based on the adaptive support weights. In addition, in terms of the computation time, the proposed method is not superior to [4] – for instance, the proposed method takes $\approx 8.422$ s for the ‘Tsukuba’ images when the number of iteration is 50 while the adaptive support-weight approach [4] takes $\approx 7.984$ s when using a $35 \times 35$-sized window with the PC equipped with 3.3-GHz i7 cpu and 4-GB memory. However, the proposed method yields slightly better performance near the depth discontinuity areas as shown in Table 2, because the support weights are iteratively adjusted while taking the visibility of each pixel into account. In addition, the proposed method requires less memory – when using similar data structures, it requires 200 Mb for the ‘Cone’ images while [4] requires about 1.8 GB although the resultant disparity maps have similar accuracy. Therefore the proposed approach is more appropriate for systems with small memory such as mobile devices than the adaptive support-weight approach [4] that requires large memory when using a large local window to deal with large textureless areas.

6 Conclusion

In this work, we have proposed a new nonlinear diffusion method for support aggregation in stereo matching. We first compute adaptive support weights based on the colour similarity and the distance between pixels. Supports are then iteratively aggregated with the support weights while adjusting the support weights according to disparities and taking the visibility of pixels into account. Experimental results show that the proposed method yields good results not only in homogeneous areas but also in depth discontinuity areas as the iteration goes on without the critical degradation of performance. However, although the
proposed method requires much less memory than using large local windows, it still requires relatively large computation power. For this reason, we are developing an efficient implementation scheme for near real-time applications of the proposed method.

7 Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2009-0065038).

8 References